Unsupervised Object Pose Classification from Short Video Sequences[1]

Liang Mei
alexmei@umich.edu
Min Sun
sunmin@umich.edu
Kevin M.Carter
kmcarter@umich.edu
Alfred O. Hero III
hero@umich.edu
Silvio Savarese
silvio@eecs.umich.edu

Department of EECS,
University of Michigan
Ann Arbor, USA

Abstract

We address the problem of recognizing the pose of an object category from video sequences capturing the object under small camera movements. This scenario is relevant in applications such as robotic object manipulation or autonomous navigation. We introduce a new algorithm where we model an object category as a collection of non parametric probability densities capturing appearance and geometrical variability within a small area of the viewing sphere for different object instances. By regarding the set of frames of the video as realizations of such probability densities, we cast the problem of object pose classification as the one of matching (i.e., comparing information divergence of) probably density functions in testing and training. Our work can be also related to statistical manifold learning. By performing dimensionality reduction on the manifold of learned PDFs, we show that the embedding in the 3D Euclidean space yield meaningful trajectories which can be parameterized by the pose coordinates on the viewing sphere, this enables an unsupervised learning procedure for pose classification. Our experimental results on both synthesized and real world data show promising results toward the goal of accurate and efficient pose classification of object categories from video sequences.

1 Introduction

Designing vision systems for enabling efficient and accurate scene interpretation is one of the greatest challenges in computer vision and related applications. In robotic manipulation, a robotic arm may need to detect and grasps objects in the scene such as a cup or book; in autonomous navigation, an unmanned vehicle may need to recognize and interpret the behavior of pedestrians and other vehicles in the environment. In all these applications, not only does one need to tackle the problem of object categorization but it is also critical to accurately estimate the pose of unknown objects in the scene: if a robotic arms wishes to grasp a mug, the system must estimate mug's pose with high degree of accuracy. While a large amount of research has been dedicated to the problem of categorizing object observed from a restricted set of views [12, 13, 19, 33], only recently a number of methods have been

[1]This research was partially supported by ARO grant W911NF-09-1-0310.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Unsupervised Object Pose Classification from Short Video Sequences	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Michigan,Department of Electrical and Computer Engineering,Ann Arbor,MI,48109**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

proposed for detecting and recognizing object classes from arbitrary view point conditions [8, 16, 17, 21, 24, 27, 28, 30, 31, 32, 34]. Critically, just a subset of these have addressed the issue of estimating the pose of an object category [21, 28, 31]. While most of the previous literature has focused on studying cues that can be extracted from a single image, in this work we use videos sequences for solving the problem of accurate pose estimation. We believe that the additional information provided by the video sequence in training and testing (that is, the temporal coherency of the object appearance as the camera moves around) plays a critical role in eliminating the inherent ambiguity in pose configurations. Unlike [3, 10, 20, 23, 25, 26, 35], however, our goal is to estimate the pose of an object instance that has not been already observed in training; thus we seek to learn object representations that enable the recognition of object poses at categorical level.

Our work starts by observing that a video sequence (portraying an object as the camera position and view point changes) can be used to parameterize a trajectory of positions on the viewing sphere, where each position corresponds to the azimuth and zenith angle coordinates describing the pose of the object (Fig.1). Our key idea is to decompose the video sequence into pockets of frames (video segments). Thus, each video segment can be associated to a location on the viewing sphere that captures the average pose within the video segments. Notice that our work is related to the large literature on manifold learning [5, 6, 11] and its application to computer vision tasks [15]. By regarding images as low-dimensional (non-linear) manifolds embedded in the high-dimensional image space, manifold learning is designed to analyze the low-dimensional structure which underlies a collection of high-dimensional data. Recent studies in statistical manifold learning [6] define information divergence as a metric of distance between probability densities and apply common dimensionality reduction techniques for visualization. Inspired by [6], we estimate probability densities using nonparametric kernel density estimation techniques and evaluate similaritues between those densities via the Kullback-Leibler divergence. Classical multidimensional scaling (cMDS) [9] can be then adopted to reconstruct the manifold in a low dimensional Euclidian space, where the pairwise KL distances are preserved through dimensionality reduction. We find that the manifold of pose trajectories forms meaningful clusters in a Euclidean embedding and enables an unsupervised learning procedure for pose estimation.

We demonstrate the recognition accuracy of the proposed algorithm on both synthesized and real datasets. Supervised classification results show that our method achieve an overall accuracy of 86.4% on a real car dataset and 85.4% on a real PC mouse dataset. Comparison with state-of-the-art spatial pyramid matching framework [14, 18] shows that our algorithm outperforms the spatial pyramid matching consistently, with a notable $10\%-20\%$ lead when the detected location of the object is corrupted by noise. We also test our unsupervised learning algorithm and obtain an accuracy of 72.1% and 57.7% for these two datasets respectively.

2 Problem Formulation

We define the problem of object pose estimation as follows. In the training stage, we are given a collection of video sequences $V = \{V^1, \ldots, V^N\}$, where V^i captures an object instance O_i. Here we assume all the object instances belong to the same object category C, and different object instances vary in shape and texture. During each video sequence, the camera moves around the object along an arbitrary trajectory on the viewing sphere. We do not assume prior information on camera movement. If we assume that the object lies at the center of the viewing sphere (Fig.1), we may describe the pose of the object as a pair of zenith and azimuth angles $q = (\theta, \varphi)$. In the testing stage, we are given a new video sequence V^t, capturing a new object instance observed around a certain viewpoint $q^t = (\theta^t, \varphi^t)$. Our goal is to estimate q^t. Note that our testing object instance does not need to appear in our

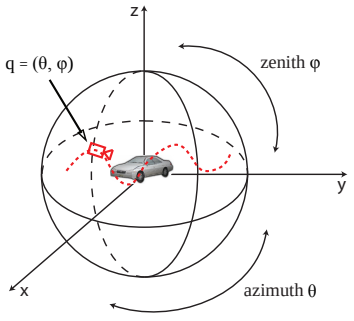


Figure 1: Pose Estimation as a pair of azimuth and zenith angles $q = (\theta, \varphi)$ on the viewing sphere. The object is assumed to lie at the center of the viewing sphere.

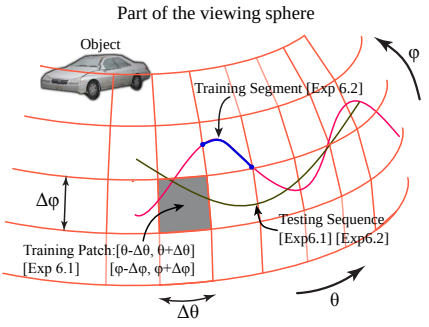


Figure 2: Trajectories showing the camera movement. Images sampled from a small segment are used for training in Exp. 5.2. Images sampled from a small patch are used for training in Exp. 5.1.

training data set, but we do assume it belongs to the same object category. Also since we do not assume any predetermined camera motion, our videos for training and testing are not necessarily taken from consistent trajectories on the viewing sphere.

In this paper we seek to solve this pose estimation problem from video segments by using statistical manifold learning techniques. We regard videos (image sets) as realizations of PDFs, where object category, shape, texture and pose are interpreted as hidden parameters. Object poses are eventually estimated in an information geometric framework, where similarities between poses are measured by information divergence between underlying PDFs.

The rest of this paper is organized as follows. A review of statistical manifold learning, information divergence theory and dimensionality reduction on statistical manifold is presented in Section 3. In Section 4, we model our object pose estimation problem in the statistical manifold learning framework and propose an algorithm to estimate pose from unseen object instances. Experimental results using our method and a benchmark experiment based on spatial pyramid matching framework [14, 18] are presented and discussed in Section 5. Finally, we conclude our paper in Section 6.

3 Statistical Manifold Learning

A manifold $\mathscr{M}$ is a locally Euclidean topological space which has a coordinate function ϕ to map every point $m \in \mathscr{M}$ to a point $p = [P_1 \dots P_d]^T \in \mathbb{R}^d$, where d is known as the dimension of $\mathscr{M}$, and $[P_1 \dots P_d]^T$ serves as a coordinate system. Statistical manifolds are manifolds of probability distributions. Define $\mathscr{M} = \{p(x|\pi)|\pi \in \Pi \subseteq \mathbb{R}^d\}$, with $p(x|\pi) \geq 0, \forall x \in \mathscr{X}$ and $\int p(x)dx = 1$. Then $\mathscr{M}$ is known as a statistical manifold on $\mathscr{X}$; π serve as a coordinate system for the manifold, and there exists a one-to-one mapping between π and $p(x|\pi)$.

3.1 Fisher Information Distance

As shown in [5], Fisher information distance is used as a metric to evaluate the divergence between probability densities. For a family of probability density functions (PDFs) $f(x;\pi_1), \dots, f(x;\pi_d)$, the Fisher information distance is defined as

$$D_F(\pi_1, \pi_2) = \min_{\pi(\cdot):\pi(0)=\pi_1, \pi(1)=\pi_2} \int_0^1 \sqrt{(\frac{d\pi}{dt})^T [I(\pi)] (\frac{d\pi}{dt})} dt \quad (1)$$

where matrix $[I(\pi)]$, known as Fisher information distance, is defined with element

$$[I(\pi)]_{ij} = \int f(X;\pi) \frac{\partial log f(X;\pi)}{\partial \pi^i} \frac{\partial log f(X;\pi)}{\partial \pi^j} dX. \quad (2)$$

Essentially, (1) amounts to the geodesic distance on manifold $\mathscr{M}$ connecting coordinates θ_1 and θ_2. When prior information regarding the parameterization of the manifold is not available, equation (1) cannot be solved explicitly. As discussed in [7], symmetric Kullback-Leibler divergence (KL-divergence) can be used to approximate the Fisher information distance. Given two PDFs p_1 and p_2, we have

$$D_{KL}(p_1,p_2)_{sym} = \int p_1(x) log \frac{p_1(x)}{p_2(x)} dx + \int p_2(x) log \frac{p_2(x)}{p_1(x)} dx \quad (3)$$

which converges to the Fisher information distance: $\sqrt{D_{KL}(p_1,p_2)_{sym}} \to D_F(p_1,p_2)$ as $p_1 \to p_2$. When p_1 and p_2 do not lie closely together on the manifold, this approximation becomes weak. Thus, we may update the distance between p_1 and p_2 by their geodesic distance, which is the sum of a series of paths connecting closely related points on the manifold. Specifically, given the collection of N probability distributions $P = \{p_1,\ldots,p_N\}$, we define an approximation of the geodesic distance for all pairs of PDFs as

$$D_G(p_1,p_2;P) = \min_{M,P} \sum_{i=1}^{M-1} D_{KL}(p_{(i)},p_{(i+1)})_{sym} \quad (4)$$

where P is the collection of PDFs on the manifold and the minimum is over all paths through the complete graph over P connecting p_1 to p_2. This geodesic distance D_G is what we finally used as an approximation of information divergence. For details, see also [7].

3.2 Manifold Clustering and Visualization

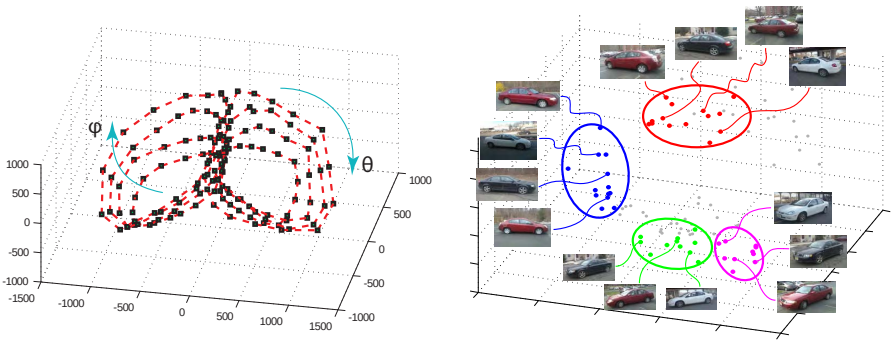
After calculation of the pairwise dissimilarity matrix of probability densities through the information divergences, we are actually building a statistical manifold. Similar PDFs form natural groups in the manifold which can be utilized for clustering and as models for unsupervised pose classification. Fig.3(b) shows 4 (of the 36) clusters obtained by applying k-means on the original manifold built using our real car dataset. Note that clusters sharing similar poses lie closer to each other.

Common multidimensional scaling techniques, such as classical Multidimensional Scaling (cMDS) [9] and Laplacian Eigenmaps [4], can be applied to the statistical manifold for the purpose of dimensionality reduction and visualization. Embedding results for a car instance from our synthesized dataset is shown in Fig 3(a), where the original video sequence is embedded as a 2D surface in a 3D Euclidean space, with object pose θ and ϕ as 2 degree of freedom. We will come back to this in Sec.4.1.

4 Classification on Statistical Manifold

Classification is nothing but estimating labels. Here we show that the general estimation problem on statistical manifold can be solved through a series of hypotheses testing, thus converting the classification problem as a detection problem.

Assume we are given sets of training data $X = \{X_1,X_2,\ldots,X_N\}$, where each data set X_i is assumed as a realization of certain PDF $P(X|\pi_i)$. In the testing stage, we are given an unseen data set X_t which we assume is generated according to $P(X|\pi_t)$, and our task is to estimate the underlying parameter π_t. To solve this problem, first we need to estimate the probability density $P(X|\pi_i)$. There are two general approaches that are usually adopted to tackle this problem. If our data come from certain parametric models (such as the Gaussian Mixture Model used in [1]), then general Maximum Likelihood techniques such as the EM algorithm



(a) (b)

Figure 3: 3(a) Embedding of estimated PDFs from a single instance of the synthesized car data (See Section 5.1). Each point in the figure corresponds to a PDF, which is estimated from images taken from a $10^\circ \times 10^\circ$ small patch on the viewing sphere (Refer to Fig.2). Trajectories in the manifold show the two main parameterizations of the learned PDFs, which corresponds to two intrinsic degrees of freedom (θ, φ) in the data. 3(b) The manifold can be naturally used to discover clusters for unsupervised pose estimation.

can be used to estimate the hidden parameters. However, in cases where prior information of the data is unknown or inaccurate, non-parametric models are used and estimated through Kernel Density Estimation (KDE). In this paper, we take the latter approach.

With the knowledge of PDFs, this estimation problem is solved by a N-ary hypothesis test, where the hypotheses are those probability densities we learned in the training stage

$$H_0 : X_t \sim P(X|\pi_0) \ldots H_N : X_t \sim P(X|\pi_N) \tag{5}$$

By Neyman-Pearson lemma [29], the optimal decision rule for this N-ary hypotheses testing problem is choosing H_i whose X_t is associated to the highest likelihood, which can be approximated by finding the hypothesis with the minimal KL-divergence. So the testing parameter π_t is estimated as

$$\pi_t = \arg\max_{\pi_i} P(X|\pi_i) \simeq \arg\min_{\pi_i} D_G(P(X|\pi_i), P(X|\pi_t)). \tag{6}$$

That is to say, we are actually doing a Nearest Neighbor classification by assigning the label of the most similar dataset in training to the testing data, where similarity is measured through information divergence.

Further more, by utilizing information divergence as a measurement of similarity between data, we can apply more sophisticated classifiers, such as the Support Vector Machine as used in [6] or Weighted Parzen Window Classifier [7] for final classification.

4.1 Object Pose Classification

By viewing images as realizations of probability distributions, as discussed in Section 2, we are able to formulate our problem of object pose classification within a statistical manifold learning framework. Specifically, assuming the object lies at the center of the viewing sphere, our observation X is generated according to

$$P(X|\mathscr{C}, T, \rho, \theta, \varphi), \tag{7}$$

where $\mathscr{C}$ is the object category; T is the texture, which captures the appearance of an object instance; ρ is the distance between the object and the camera, which affects the object scale; θ and φ are azimuth and zenith angles representing the viewpoint, respectively. By assuming all the objects belong to the same category, and ρ is fixed (small scale variations can be

accommodated by normalizing the object bounding box to unit length, as we will discuss in detail in Section 5), the probability density function (7) can be rewritten as

$$P(X|\mathscr{C},T,\rho,\theta,\varphi) = P(X|T,\theta,\varphi) \tag{8}$$

Note here the observation vector X can be represented in different ways. While pixel intensity value is the most straightforward one, various pre-processing techniques (such as the Canny edge detector) and feature descriptors (such as SIFT [27]) can also be used. As we shall see in Section 5.2, edge based features help make our algorithm more robust in discriminating object poses.

Suppose we are given a video sequence $\mathscr{V}^i$ capturing the object instance i as view point $q = (\theta,\varphi)$ varies on the viewing sphere. We divide the video into segments of length K, and regard frames in segment j ($j = 1,2,\ldots,\lceil N^i/K\rceil$, where N^i is the number of frames in $\mathscr{V}^i$) as generated according to

$$P^i_j = P(X|T = t^i, \theta \in [\theta_j - \Delta\theta_j, \theta_j + \Delta\theta_j], \varphi \in [\varphi_j - \Delta\varphi_j, \varphi_j + \Delta\varphi_j]) \tag{9}$$

where $[\theta_j - \Delta\theta_j, \theta_j + \Delta\theta_j], [\varphi_j - \Delta\varphi_j, \varphi_j + \Delta\varphi_j]$ defines the angular support of the segment j on the viewing sphere (Fig.2).

The PDFs (9) are estimated through KDE. Then the KL-divergence between all possible pairs of PDFs are calculated. We use classical multidimensional scaling (cMDS) to reduce dimensionality and reconstruct the statistical manifold. This gives rise to a manifold which consists of $\sum_i N^i/K$ points, where each point corresponds to a probability density (9). Fig.3(a) shows an example of the embedded PDFs from the synthesized car dataset (Section 5.1) in a 3D space. Each point in the figure is estimated from images taken from a $10^\circ \times 10^\circ$ small patch on the viewing sphere (E.g. See Fig.2). Trajectories in the manifold in Fig.3(a) show the two main parameterizations of the learned probability models, which corresponds to two intrinsic degrees of freedom (θ,φ) in the data.

In testing, we are given a small video sequence V^t of a new object instance O_t from an unknown viewpoint, and our goal is to estimate the viewpoint $q = (\theta_t,\varphi_t)$ of the test video. We explored two classification schemes. By following the Neyman-Pearson lemma and the hypothesis testing scheme, we apply a nearest neighbor classifier to estimate object pose in the video sequence. A weighted Parzen window predictor was also tested and shown to yield higher classification accuracy.

5 Experiments

In this section, we show that our algorithm is able to successfully recognize the pose of an object given a short video sequence capturing the object under small view point changes.

5.1 Pose classification with synthesized data

We first conduct experiments on a synthesized car dataset, which contains ten 3D car models mapped with texture from real photographs - such photographs are taken from the database presented in [27]. By changing viewpoint $\theta \sim [0^\circ, 360^\circ]$, $\Delta\theta = 1^\circ$, $\phi \sim [0^\circ, 40^\circ]$, $\Delta\phi = 1^\circ$, we generate $360 \times 40 = 14400$ images for each car instance.

A leave-one-out cross validation scheme is adopted on those 10 car instances. Test object instances are never used in training for estimating relevant PDFs. During training, a PDF as in (9) for instance i is estimated by considering a set of 10×10 images associated to a small patch on the viewing sphere, defined as $\theta = [\theta_0 - \Delta\theta, \theta_0 + \Delta\theta], \phi = [\phi_0 - \Delta\phi, \phi_0 + \Delta\phi]$ (See Fig.2). By choosing $\Delta\theta = 10^\circ$ and $\Delta\phi = 10^\circ$, we obtain $36 \times 4 \times 9 = 1296$ hypotheses. A test dataset is generated by taking image samples along a randomly chosen curve segment on the viewing sphere, which mimics the behavior of a moving camera (See Fig.2). Here the

Avg. accuracy = 82.23%

	F	FL	L	BL	B	BR	R	FR
F	.62	.	.	.	.18	.	.	.16
FL	.11	.81	.04	.	.02	.03	.	.
L	.	.03	.94	.02	.	.	.	.
BL	.	.	.08	.82	.	.	.	.11
B	.25	.	.	.12	.58	.	.	.05
BR	.04	.04	.	.	.11	.79	.03	.
R	.	.	.	.	.	.05	.93	.02
FR	.	.	.	.06	.	.	.06	.88

Avg. accuracy = 76.20%

	10	20	30	40
10	.84	.16	.	.
20	.14	.76	.1	.
30	.01	.16	.72	.11
40	.	.	.25	.75

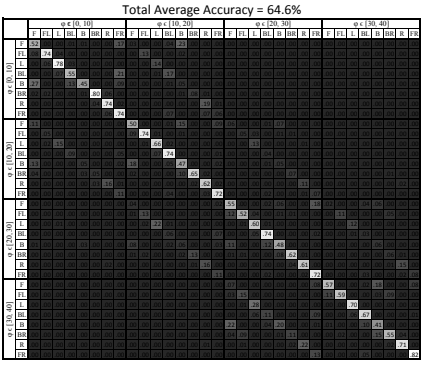


Figure 4: Confusion table reports azimuth pose (**Left**), zenith pose (**Middle**) and joint pose estimation (**Right**) for the synthesized car dataset.

length of the segment is 10 frames. Rasterized image pixel values are used to represent the observation vector X and a 1 nearest neighbor classifier is adopted for final classification.

To avoid having the bounding box shape contaminate our classification result (Frontal view of a car tends to have a smaller bounding box than the side view), images are normalized so as to make the bounding box a unit square. Fig.4 reports a summary of the estimation accuracy. We discretize the viewing sphere into 8 azimuth regions (Front, Front-Right, Right, Back-Right, Back, Back-Left, Left, Front-Left) and 4 zenith regions ($[0°,10°]$, $[10°,20°]$, $[20°,30°]$, $[30°,40°]$) for calculating the final confusion table. As shown in the figure, we achieve an average performance of 82.23% in estimating the azimuth pose and 76.20% in estimating the zenith pose. Joint estimation of θ and φ achieves an overall accuracy of 64.6%, where random guess accuracy is only $1/32 = 3.12\%$.

5.2 Pose classification with real data

In this experiment we test our algorithm on a real world dataset comprising 4 car instances and 5 PC mouse instances captured by a hand held low resolution camera; the camera trajectory covers different locations on the viewing sphere following a semi-sinusoidal trace (Fig.2). This trajectory mimics the behavior of a person observing an object - moving around the object and raising/lowering the observation point slowly. Bounding box for the object is assumed in training and testing. This assumption is reasonable in scenarios where objects are tracked or detected using off-the-shelf object detectors. To make our experiments closer to real situations, where accurate bounding box is rarely available, independent Gaussian noises are added to the top-left and bottom-right coordinates of the ground truth bounding box. Noise level is controlled by setting standard deviation of the Gaussian distribution as a function (percentage) of the width/height of the bounding box. Examples of frames from our dataset along with bounding boxes are shown in Fig.5.

As opposed to the experiment with synthesized dataset, where each PDF Eq.(9) is estimated from a (2D) patch on viewing sphere defined by $[\Delta\theta, \Delta\varphi]$, our hypotheses are now estimated on images belonging to small (1D) trajectory segments on the viewing sphere; trajectory segments are obtained by dividing the video sequence into short segments of K frames. Testing images are obtained in a similar way. In our experiment, we use $K = 10$.

Another major difference is that the synthesized images have blank backgrounds, while in real photographs objects lie in cluttered environments. Since accurate object segmentations are rarely available in real world situations, it is very important that our proposed pose estimation algorithm be robust to background noise and clutter.

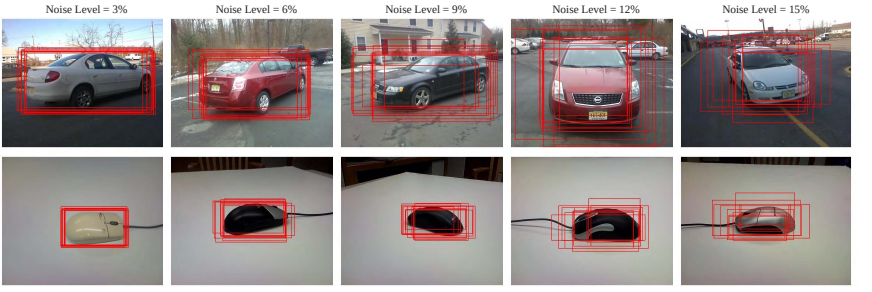


Figure 5: Some frames from the car and mouse dataset we used in the experiments, along with bounding boxes localizing the object. Note the intra-class variability (appearance difference) within each category (Modeled as texture t^i). Different degrees of noise are added to the ground truth bounding boxes. See text for details.

Image representation. We tested several methods for representing the information extracted from the images: raw pixel intensity values (image+intensity: II), SIFT descriptors computed on uniformly sub-sampled pixel locations (image+SIFT: IS), edge maps generated by Canny edge detector from original images (edge+intensity: EI), and SIFT descriptors computed on uniformly sub-sampled edge maps (edge+SIFT: ES). For compuatational efficiency and tractability, data from all these representations is pre-processed by PCA and the first 50 principle components are fed to KDE to estimate the actual PDFs (experimental results indicate that our system produces stable and consistent results if more than 50 principle components are used). In our experiments, a weighted Parzen window classifier is used to estimate the pose label, where the Parzen window size is chosen empirically.

For a given noise level, we generate 10 realizations of the noisy bounding boxes and repeat classification 10 times for each image representation. This scheme helps average out performance variability due to noise and lead to more stable quantitative evaluations. Relevant average accuracies (with standard deviations) are reported as an assessment of the performance. As shown in Fig.6(a) and 6(d), our method (using ES representation) consistently yields the highest average accuracy across all tested noise levels (with an accuracy of 86.4% for the car class and 85.4% for the mouse class at 3% noise). An interesting observation is that by using a representation based on edges, the pose recognition accuracy jumps from less than 60% (II) to up to 80% (EI) on both datasets. These results indicate that edges lead to highly discriminative capabilities in our manifold learning framework.

Video segment length. Fig.6(b) and 6(e) show the performance of our algorithm as a function of the number of frames K used for tranining/testing in each video segment. As shown in these figures, the recognition accuracy is very low when K is small, which is simply because there is not enough data for estimating the PDFs. As K gets larger ($K \sim [1-6]$), the PDFs are estimated more accurately and the performance improves significantly. This suggests that, with more frames, the contribution of noise becomes less significant and that patterns of features start emerging and becoming statistically significant. And after a certain threshold ($K > 10$), the performance becomes stable.

Number of training instances. Fig.6(c) and 6(f) summarize the recognition accuracy as a function of the number of training instances. The performance improves as more instances are used in training, indicating that our algorithm has promising generalization power.

Unsupervised pose estimation. We also demonstrate the power of using our method for unsupervised pose estimation. In this experiment we first use k-means to cluster the statistical manifold (which is build on the training data only) with number of clusters $C = 36$, and assign a unique pose label to each cluster. Then we use a Parzen window classifier to

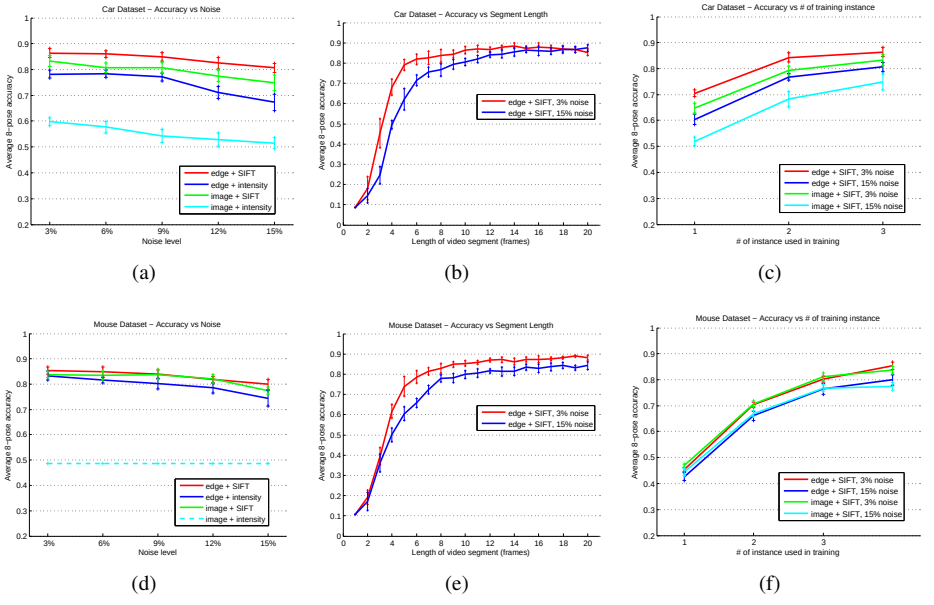


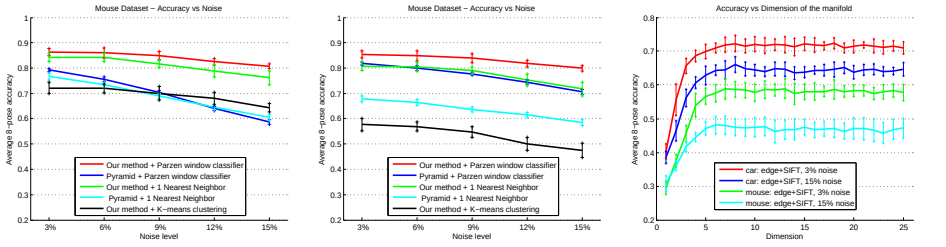
Figure 6: (**Left column**) 8-Pose classification accuracy at different noise levels. Note that our proposed method is robust to different degree of noise applied to the bounding box location and size, while the performance of pyramid matching framework drops dramatically when the bounding box is inaccurate. (**Mid column**) Classification accuracy as a function of length of video segment used for training and testing. (**Right column**) Classification accuracy as a function of number of instances used for training. (**Top row**) Experimental results with the car dataset. (**Bottom row**) Experimental results with the mouse dataset.

predict the labels of the testing video segment based on the estimated labels. Finally, we compare the predicted labels of the testing images to the ground truth and report the accuracy. We show samples of clusters in the manifold in Fig.3(b) and the recognition accuracy in Fig.7 as function of number of dimensions of the reconstructed manifold.

Comparison with [18]. As a baseline experiment, we applied the spatial pyramid matching scheme[18] on our car and mouse datasets and formulate the pose estimation as a single frame classification problem. We again adopt the leave-one-out scheme and use all the video frames for training/testing. We set the dictionary size to be 100 and calculate level 3 spatial pyramids on raw images, followed by a 1 nearest neighbor / Parzen window classifier. As shown in Fig.7, our method performs better than the pyramid matching baseline for both classifiers. Note that our algorithm tends to be more robust to noise compared to pyramid matching. As the noise level increases from 3% to 15%, our method (using a ES representation with the Parzen window classifier) outperforms pyramid matching on both dataset up to 20%.

6 Conclusion

We tackled the problem of estimating the pose of an object category from a video sequence portraying the object under small camera movements. We introduced a new algorithm that models an object category as a collection of non-parametric probability density functions capturing appearance and geometrical variability as the camera moves around the object. The problem of object pose classification is tackled by measuring the information divergence



(a) (b) (c)

Figure 7: Comparison of our method with the spatial pyramid matching scheme for the car (7(a)) and mouse (7(b)) dataset. (7(c))Unsupervised classification accuracy as a function of number of dimensions of the reconstructed manifold.

of the probably density functions in testing and training. The key advantage of this algorithm with respect to competing methods for pose classification is that no pose labeling is required in training. We demonstrated that our algorithm can successfully classify the pose of unseen instances of cars and PC mouses observed from a short period of time using a hand held low resolution camera. We believe this work represents a promising step forward for solving the challenging and yet fairly unexplored problem of pose classification from video imagery.

References

[1] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, pages 581–588, 2005.

[2] Gregory A. Babich and Octavia I. Camps. Weighted parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 567–570, 1996.

[3] ArslanA. Basharat, Y. Zhai, and M. Shah. Content based video matching using spatiotemporal volumes. *Journal of Computer Vision and Image Understanding*, 2008.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.

[5] K. Carter, C. Kim, R. Raich, and A. Hero. Information preserving embeddings for discrimination. In *Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, Jan. 2009.

[6] K. Carter, R. Raich, W. Finn, and A. Hero. Fine: Fisher information nonparametric embedding. *IEEE Transactions on Pattern Recognition and Machine Learning*, 2009.

[7] Kevin M. Carter. *Dimensionality Reduction on Statistical Manifolds*. PhD thesis, University of Michigan, Ann Arbor, Michigan, 2009.

[8] H. Chiu, L. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. In *CVPR*, 2007.

[9] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, 2000. second edition.

[10] E. Delponte, N. Noceti, F. Odone, and A. Verri. The importance of continuous views for real-time 3d object recognition. In *ICCV07 Workshop on 3D Representation for Recognition*, 2007.

[11] P. Dollár, V. Rabaud, and S. Belongie. Learning to traverse image manifolds. In *NIPS*, Dec. 2006.

[12] L. Fei-Fei, R. Fergus, and A. Torralba. Recognizing and learning object categories. In *CVPR Short Course*, 2007.

[13] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[14] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[15] J. Ham, I. Ahn, and D. Lee. Learning a manifold-constrained map between image sets: applications to matching and pose estimation. In *CVPR*, pages 817– 824, Jun. 2006.

[16] D. Hoeim, C. Rother, and J.Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *CVPR*, 2007.

[17] A. Kushal, C. Schmid, and J. Ponce. Flexible object models for category-level 3d object recognition. In *CVPR*, 2007.

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[19] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Proceedings of the Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004.

[20] M. Leordeanu and R. Collins. Unsupervised learning of object features from video sequences. In *CVPR*, pages 1142–1149, Jun. 2005.

[21] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, Jun. 2008.

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2:91–110, 2004.

[23] G. Michael and B. Horst. Object recognition based on local feature trajectories. In *1st Cognitive Vision Workshop*, 2005.

[24] H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[25] H. Najafi, Y. Genc, and N. Navab. Fusion of 3d and appearance models for fast object detection and pose estimation. In *7th Asian Conference on Computer Vision (ACCV)*, 2006.

[26] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, pages 231–259, 2006.

[27] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. *ICCV*, pages 1–8, 2007.

[28] S. Savarese and L. Fei-Fei. View synthesis for recognizing unseen poses of object classes. In *ECCV*, 2008.

[29] L. Scharf and C. Demeure. *Statistical signal processing*. Addison-Wesley, 1991.

[30] G. Shakhnarovich, J. W. Fisher, and Trevor Darrell. Face recognition from long-term observations. In *ECCV*, pages 851–868, 2002.

[31] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3d object classes. In *CVPR*, 2009.

[32] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, volume 2, pages 1589–1596, 2006.

[33] A. Torralba, K. Murphy, and W. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004.

[34] P. Yan, D. Khan, and M. Shah. 3d model based object class detection in an arbitrary view. In *ICCV*, 2007.

[35] Z. Yin and R. Collins. On-the-fly object modeling while tracking. In *CVPR*, Jun. 2007.